Cyclistic Rider Comparison

Ana Faria
2020 Q1

Introduction

This notebook contains my first case study for my Google's Data Analytics Professional Certificate. This case study requires the analyst to follow to the steps of the data analysis process (ask, prepare, process, analyze, share,and act) for a data set that represents a fictional company, Cyclistic Bike-Share.

Case Study Business Task - Ask

For this case study our objective business task is to find how bike usage differs from annual membership holders to casual riders who use the Cyclistic bike service in order to understand the best marketing strategy to convert casual riders into membership holders, which will maximize profits. This project looks to find the main differences between casual riders and member riders.

Package downloads

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(tidyverse) #helps wrangle data
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.
0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
```

```
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(lubridate) #helps wrangle data attributes
library(ggplot2) #for data visualization
setwd("/cloud/project") #sets your working directory to simplify calls to dat
alib
```

Data Collection - Prepare

For this case study we are using a public dataset provided by Divvy Bikes as a proxy for our fictitious company's ridership history. The dataset covers Q1 2020 service usage, including type of bike used, date time ride started, date time ride ended, start station name and id, end station name and id, start latitude, start longitude, end latitude, and end longitude. Data Privacy laws limit the amount of personally identifiable data that could lead us to relevant information regarding casual rider's bike usage, including if they live close to a Cyclistic service area or if they purchase multiple one day passes.

Reading the data

In this section I uploaded the data into R Studio.

```
q1_2020 <- read_csv("Divvy_Trips_2020_Q1.csv")

## Rows: 426887 Columns: 13
## ── Column specification ────────────────────────────────────────────────────
──────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, me
mb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat,
e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this m
essage.
```

Wrangle Data and Combine Into One File - Process

Dataset column names

Below I was able to pull all the variables available in the data provided.

```
colnames(q1_2020)

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
```

```
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

Data types per column

This output determines the data type for each variable. This will help determine if there is any variable's data type that needs to be modified for further analysis.

```
str(q1_2020)
```

```
## spc_tbl_ [426,887 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021
" "789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_
bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2
020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2
020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St
& Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 .
..
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport A
ve & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ..
.
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100
...
##  $ start_lat         : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member"
...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Removing columns on latitude and longitude

In this data chunk we eliminated latitude and longitude variables in order to better organize our data set. For the purpose of this analysis we do not need both of this variables.

```
q1_2020 <- q1_2020 %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

Data cleanup and preparation for analysis

In this data chunk, we first verified that all relevant variables are part of the modified data set. We then determine that there are 426,887 observations in our data set, described by 9 variables. We also were able to determine statistical information regarding our data, specifically we pulled the mean, mode, median, max and min values for each variable. For numerical variables, we were also able to pull the normal distribution spread for each variable.

```
colnames(q1_2020)  #List of column names
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "start_station_id"
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

```
nrow(q1_2020)  #How many rows are in data frame?
```

```
## [1] 426887
```

```
dim(q1_2020)  #Dimensions of the data frame?
```

```
## [1] 426887      9
```

```
head(q1_2020)  #See the first 6 rows of data frame.  Also tail(qs_raw)
```

```
## # A tibble: 6 × 9
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 EACB19130B0CDA4A docked_bike   2020-01-21 20:06:59 2020-01-21 20:14:30
## 2 8FED874C809DC021 docked_bike   2020-01-30 14:22:39 2020-01-30 14:26:22
## 3 789F3C21E472CA96 docked_bike   2020-01-09 19:29:26 2020-01-09 19:32:17
## 4 C9A388DAC6ABF313 docked_bike   2020-01-06 16:17:07 2020-01-06 16:25:56
## 5 943BC3CBECCFD662 docked_bike   2020-01-30 08:37:16 2020-01-30 08:42:48
## 6 6D9C8A6938165C11 docked_bike   2020-01-10 12:33:05 2020-01-10 12:37:54
## # ℹ 5 more variables: start_station_name <chr>, start_station_id <dbl>,
## #   end_station_name <chr>, end_station_id <dbl>, member_casual <chr>
```

```
str(q1_2020)  #See list of columns and data types (numeric, character, etc)
```

```
## tibble [426,887 × 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021
" "789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_
bike" "docked_bike" ...
```

```
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2
020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2
020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St
& Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 .
..
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport A
ve & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ..
.
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100
...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member"
...
```

```
summary(q1_2020)  #Statistical summary of data. Mainly for numerics
```

```
##    ride_id          rideable_type        started_at
##  Length:426887      Length:426887      Min.   :2020-01-01 00:04:44.00
##  Class :character   Class :character   1st Qu.:2020-01-24 14:03:26.00
##  Mode  :character   Mode  :character   Median :2020-02-17 05:01:27.00
##                                        Mean   :2020-02-14 01:23:18.51
##                                        3rd Qu.:2020-03-05 15:08:13.50
##                                        Max.   :2020-03-31 23:51:34.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2020-01-01 00:10:54.00   Length:426887      Min.   :  2.0
##  1st Qu.:2020-01-24 14:21:24.50   Class :character   1st Qu.: 77.0
##  Median :2020-02-17 05:48:58.00   Mode  :character   Median :176.0
##  Mean   :2020-02-14 01:45:25.43                      Mean   :209.8
##  3rd Qu.:2020-03-05 15:27:54.00                      3rd Qu.:298.0
##  Max.   :2020-05-19 20:10:34.00                      Max.   :675.0
##
##  end_station_name   end_station_id  member_casual
##  Length:426887      Min.   :  2.0   Length:426887
##  Class :character   1st Qu.: 77.0   Class :character
##  Mode  :character   Median :175.0   Mode  :character
##                     Mean   :209.3
##                     3rd Qu.:297.0
##                     Max.   :675.0
##                     NA's   :1
```

Problems in data set that needs fixing before analysis

After processing the data, we determined a few errors that must be fixed before conducting our analysis.

- The data can only be aggregated at the ride-level, which is too granular. We will want to add some additional columns of data – such as day, month, year – that provide additional opportunities to aggregate the data.

- We will want to add a calculated field for length of ride since the 2020 Q1 data did not have the "tripduration" column. We will add "ride_length" to the entire data frame for consistency.
- There are some rides where tripduration shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. We will want to delete these rides.

```
q1_2020$date <- as.Date(q1_2020$started_at) #The default format is yyyy-mm-dd
q1_2020$month <- format(as.Date(q1_2020$date), "%m")
q1_2020$day <- format(as.Date(q1_2020$date), "%d")
q1_2020$year <- format(as.Date(q1_2020$date), "%Y")
q1_2020$day_of_week <- format(as.Date(q1_2020$date), "%A")
```

Calculation for ride_length in seconds

In this data chink, we calculated the length of each ride recorded in Q1 2020. Ride_length is the difference between the ended_at and started_at variables.

```
q1_2020$ride_length <- difftime(q1_2020$ended_at,q1_2020$started_at)

str(q1_2020)
```

```
## tibble [426,887 × 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021
" "789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_
bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2
020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2
020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St
& Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 .
..
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport A
ve & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ..
.
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100
...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member"
...
##  $ date              : Date[1:426887], format: "2020-01-21" "2020-01-30" .
..
##  $ month             : chr [1:426887] "01" "01" "01" "01" ...
##  $ day               : chr [1:426887] "21" "30" "09" "06" ...
##  $ year              : chr [1:426887] "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr [1:426887] "Tuesday" "Thursday" "Thursday" "Mon
day" ...
```

```
##  $ ride_length       : 'difftime' num [1:426887] 451 223 171 529 ...
##   ..- attr(*, "units")= chr "secs"
```

Converting ride_length factor into numeric

In this data chunk we formatted our ride_length variable into a numeric data type in order to be able to pull descriptive statistic from our findings.

```
is.factor(q1_2020$ride_length)

## [1] FALSE

q1_2020$ride_length <- as.numeric(as.character(q1_2020$ride_length))
is.numeric(q1_2020$ride_length)

## [1] TRUE
```

Removal of bad data for new data set all_trips_v2

In this data chunk, we got rid of the observation that include moving the bikes to headquarters and ride durations incorrectly inputted as negative numbers.

```
all_trips_v2 <- q1_2020[!(q1_2020$start_station_name == "HQ QR" | q1_2020$rid
e_length<0),]
```

Descriptive Analysis - Analysis

Descriptive analysis on ride_length, all figures in seconds

The following shows a statistical summary of the new variable ride_length for all observations.

```
mean(all_trips_v2$ride_length) # average ride length in seconds

## [1] 1338.697

median(all_trips_v2$ride_length) #midpoint number in the ascending array of r
ide lengths in seconds

## [1] 555

max(all_trips_v2$ride_length) #longest ride in seconds

## [1] 9387024

min(all_trips_v2$ride_length) #shortest ride in seconds

## [1] 1
```

Descriptive Analysis Summary For ride_length

Note to self: Should add a normal distribution graph here

In this data chunk we can observe that the ride_length is right skewed as the mean of 1339 seconds is greater than the median of 555 seconds per ride. These means that the most riders tend to do longer than average rides rather than shorter than the average of 1,339 seconds of bike riding.

```
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     334     555    1339     955 9387024
```

Member vs. Casual Users Comparison

Through this comparison we determined that on average casual riders have longer rides than members. That can be explained by the incentive of using a day pass in order to make the best out of their investment. Members seem to observe their investment into bike riding as sunk cost, hence they may be less incentivized to take longer rides.

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
#average ride length in seconds
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                6230.7734
## 2                     member                 760.6287
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median
) #mid-point of ride length in seconds
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                     1389
## 2                     member                      515
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max) #
longest ride in seconds
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  9387024
## 2                     member                  5627611
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min) #
shortest ride in seconds
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        2
## 2                     member                        1
```

Member vs. Casual Users Rides per Day of the week

In this data chunk, we analyzed each riders riding patterns per day of the week. In our result we can observe that casual riders tend to ride the most on Thursdays, while members ride on Sunday's the most. Upon closer inspection, we determined that although rides are longer on Thursdays for casual riders, the number of bike riding instances are higher on Sundays. For members, even though the rides on Sunday last longer on average, they use the service the most on Tuesdays.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunda
y", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")) # to
order days of the week

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v
2$day_of_week, FUN = mean) #average ride length per day of the week for each
type of user
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_l
ength
## 1                      casual                   Sunday                5710
.5665
## 2                      member                   Sunday                 949
.3401
## 3                      casual                   Monday                5818
.3439
## 4                      member                   Monday                 778
.6286
## 5                      casual                  Tuesday                5832
.3594
## 6                      member                  Tuesday                 692
.0323
## 7                      casual                Wednesday                5132
.6226
## 8                      member                Wednesday                 699
.5471
## 9                      casual                 Thursday                8744
.6574
## 10                     member                 Thursday                 693
.2325
## 11                     casual                   Friday                7907
.8883
## 12                     member                   Friday                 757
.3241
## 13                     casual                 Saturday                6017
.1560
## 14                     member                 Saturday                 929
.9892
```

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday fiel
```

```
d using wday()
    group_by(member_casual, weekday) %>% #groups by user type and weekday
    summarise(number_of_rides = n() #calculates the number of rides and avera
ge duration
    ,average_duration = mean(ride_length)) %>%  #calculates the average durat
ion
  arrange(member_casual, weekday)   #sorts
```

```
## `summarise()` has grouped output by 'member_casual'. You can override usin
g the
## `.groups` argument.
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun               14886            5711.
##  2 casual        Mon                3699            5818.
##  3 casual        Tue                4583            5832.
##  4 casual        Wed                5201            5133.
##  5 casual        Thu                4227            8745.
##  6 casual        Fri                4638            7908.
##  7 casual        Sat                7480            6017.
##  8 member        Sun               35964             949.
##  9 member        Mon               61923             779.
## 10 member        Tue               69697             692.
## 11 member        Wed               63977             700.
## 12 member        Thu               61245             693.
## 13 member        Fri               55496             757.
## 14 member        Sat               30104             930.
```

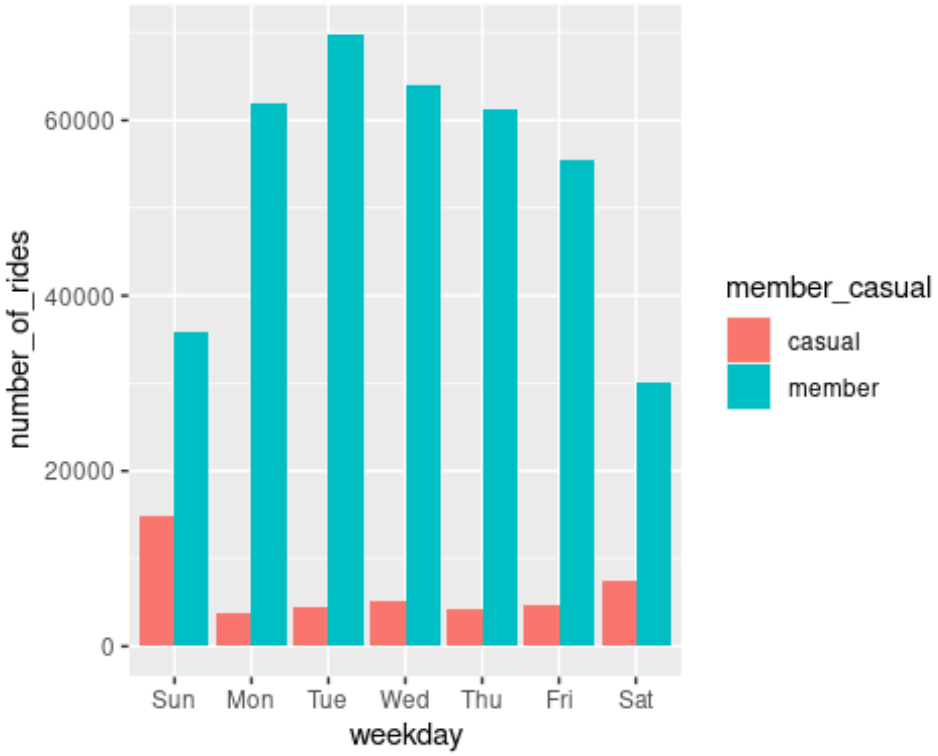
Data Visualizations - Share

Number of Rides each Day of the Week per Rider Type

In the first visualization below, we observe our previous finding, where members use the service the most on Tuesdays, while casual riders use the service the most on Sundays. The second visualization shows how even though members use the service the most amount of times, casual riders use the service for longer rides.

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

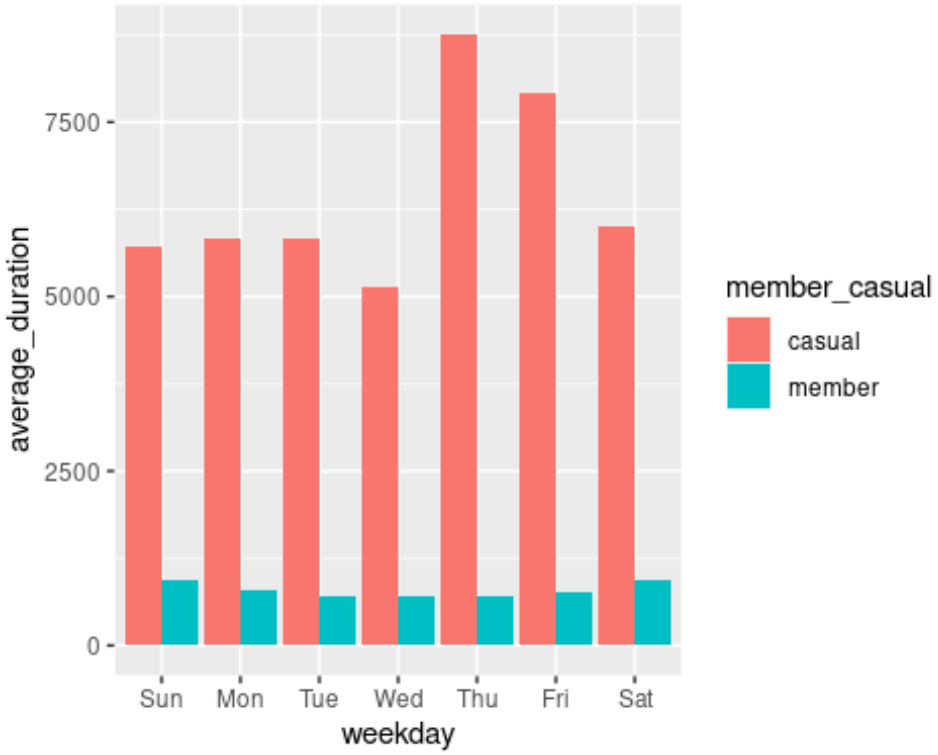
```
## `summarise()` has grouped output by 'member_casual'. You can override usin
g the
## `.groups` argument.
```



Average Ride each Day of the Week per Rider Type

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

```
## `summarise()` has grouped output by 'member_casual'. You can override usin
g the
## `.groups` argument.
```



Conclusion - Act

In this analysis we found that the main difference between casual riders and annual members is that casual riders travel for a longer time than members, this may be due to an incentive to travel longer as they are making the payment and immediately using the bike service. On the contrary, members usually prepay their service, hence they may be less incentivize to user the bike for a longer time. On the other hand, casual riders use the service during the weekend, thus it would be imperative to include marketing incentives catered for weekend activities. The main take away through this analysis annual members use the biking service for day to day use, while casual riders use it for one-time occasions. For further analysis we recommend observing which areas are more commonly used by casual users, we could incentivize them to use the service more regularly though consecutive weekends activities that may incentivize them to use the service regularly.